Amendment to the Drawings:

The attached sheet of drawings includes a change to FIG. 2 to replace a reference that was used for two different components.

Attachments: Annotated sheet showing changes between originally filed FIG. 2 and FIG. 2 as filed January 23, 2004;
Replacement sheet for FIG. 2; and
Annotated sheet showing changes between FIG. 2 filed January 23, 2004 and the replacement FIG. 2 filed herewith.

REMARKS

In paragraphs 2, 3, and 4 of the Office action, the examiner identifies inconsistencies between the specification and figure 2 as filed. It is believed that all of the inconsistencies identified by the examiner were corrected at the time that formal drawings were submitted on January 23, 2004 in response to a Notice to File Corrected Application Papers issued on January 20, 2004. For the convenience of the examiner, a copy of the as filed figure 2 marked in red to shown the changes between the as filed figure 2 and the formal figure 2 filed on January 23, 2004 is enclosed.

In reviewing the formal drawings filed on January 23, 2004, it was determined that the reference number 46 was used twice in figure 2, once for the multiplexer and once for the host memory access port. Both the specification and the figure have been amended to use the reference number 48 for the host memory access port. If the examiner is of the belief that the drawings still require correction, the examiner is invited to contact the undersigned attorney so that any remaining problems with the figures can be immediately addressed.

In response to paragraph 6 of the Office action, the Abstract has been amended as suggested by the examiner.

In response to paragraph 7 of the Office action, the specification has been amended such that the first occurrence of MMX contains a plain text description of that acronym.

In response to paragraph 9 of the Office action, independent claims 1, 2, 6, 9, and 15 have been amended to provide more explanation regarding the meaning of the variables in the claims. If the examiner is of the belief that the claims are still indefinite, the examiner is invited to contact the undersigned attorney so that any remaining issues under 35 U.S.C. § 112 can be immediately addressed.

In response to paragraph 11 of the Office action, all of the independent claims, claims 1, 2, 6, 9, and 15, have been amended to recite that the interleave pattern or the stored table is used to distribute tasks to a plurality of processing elements to balance the workload across the plurality of processing elements. Support for that amendment can be found in paragraph [0050] of the application as filed which provides:

> **[0050]** This invention is particularly useful in combination with load balancing methods. For load balancing, there is a requirement to distribute T tasks across P PEs or, more generally, P processors. This would give a mean number of tasks M = T/P on each processor. However, in general T/P is not an integer. To preserve the number of tasks T, some processors will be assigned A tasks and some processors will be assigned B tasks, where A = truncated (T/P), and B = A + 1, and A and B are integers. If the number of processors with A tasks is X, then T = A.X + B.(P – X).

Because all of the independent claims are now limited to a particular use of the recited methods, i.e., redistribution of tasks across a plurality of processing elements for load balancing, each of the independent claims recites a useful, tangible, and concrete result.

In paragraph 13 of the Office action, claims 1-15 stand rejected under 35 U.S.C. § 103 as being unpatentable over Poeppelman (U.S. Patent No. 6,617,985) in view of Cornelius et al. (U.S. Patent No. 6,363,152) and Stephen (U.S. Patent No. 6,329,935). Applicant respectfully traverses that rejection.

With respect to claim 1, it is the examiner's position that Poeppelman teaches the "invention substantially as claimed." Applicant disagrees with that conclusion. Although it is true that if a word search is performed on Poeppelman, the terms segment A, segment B, and interleave can be found, the relevance of Poeppelman to the claimed invention is superficial at best.

Poeppelman is not directed to a method of generating an interleave pattern, and certainly not to an interleave pattern of n lots of A and lots of B. Poeppelman is directed to the following subject matter:

> The present invention concerns a method for generating constraint codes in a stream of data having a plurality of multi-bit source words, comprising the steps of (A) checking a sequence portion of the multi-bit source words for one or more constraint violations and (B) if no constraint violations are detected, modifying a predetermined portion of each of the multi-bit source words to generate a plurality of corresponding multi-bit code words configured to prevent the constraint violations of the sequence portions across an adjacent two of the multi-bit code words. Column 1, lines 24-33.

One of the constraints that is enforced is an interleave constraint, an example of which is described in column 2, lines 60-63:

> An example of an interleave (I) constraint violation of 13 consecutive zeroes on an even or odd interleave (illustrated as the underlined zeros) may be illustrated as: 1110101011101010101010101010101010101011101

The portion of Poeppelman cited by the examiner has to do with sequence replacement when there is a constraint violation, not the creation of an interleave pattern. The sequence replacement step 310 is described in the paragraph bridging columns 3 and 4 as follows:

> Referring to FIG. 4, an example of sequence replacement step 310 is shown. Sequence replacement 310 may be implemented if the source word 302 contains a first type (e.g., an internal G or I) of constraint violation. The sequence replacement step 310 may generate a partially coded word 312. The SRM (e.g., 0011) may be implemented (inserted) as the suffix (e.g., the highest order bits or predetermined location) of the partially coded word 312. The first 12 bits of the constraint violation are generally removed from the source word 302. The higher order bit (e.g,. A) of the source word 302 may be appended to the SRM (e.g., 0011) and preceded by a 5-bit pointer (e.g., the bits PPPPP)that may indicate where in the source word 302 the first bit was removed and a type bit (e.g., T, where G=0 and I=1) that indicate the type of constraint violation. The removal of the constraint violation and the insertion of the position and type indicator bits may generate a 38-bit word as shown. In the example shown, a global violation starts at location 14. Therefore, position bits PPPPP=01110 (e.g., 14) and type bit T=0. The lower order bits (e.g., B) generally are mapped one-for-one. The pointer PPPPP and the type indicator T may be inserted at any appropriate predetermined position (e.g., bit location) in the partially coded word 312 to meet the design criteria of a particular application.

It is thus seen that the primary reference is substantively dissimilar from the subject matter of the independent claims. Additionally, now that the independent claims have been limited to redistribution of tasks for purposes of load balancing, it is respectfully submitted that a person of ordinary skill in the art could not reasonably be expected to look to art related to methods and apparatus for implementing constraint codes with low error rate propagation. For these reasons, it is respectfully submitted that the obviousness rejection of claim 1 should be withdrawn.

In paragraph 15 of the Office action, the examiner notes certain shortcomings in the primary reference. Namely, that "Poeppelman does not explicitly disclose of replacing the values of n in correspondence to a key." The examiner looks to Cornelius et al. for the missing teachings. Cornelius et al. discloses the following:

> The methods for transmitting and receiving encrypted digital messages or storing and retrieving encrypted data utilize the functional blocks shown in FIG. 1 and FIG. 2, to accomplish the encryption and decryption of a signal or a stored data representation. Referring to FIG. 1, [a] random sequence of numbers is generated which serves as the one time pad signal 24 and is combined with the input data 22 by a one time pad encoder 14 to produce an encrypted data stream. The one time pad signal 24 is encrypted by means of the above mentioned DES, RSA or

> other "short key" technique by a pad encrypter 16 to produce an encrypted pad and then the encrypted pad and the encrypted data stream are interleaved and output as a single data stream, with the interleave selection made by the multiplexer 20, controlled by the unencrypted one time pad signal 24. Column 4, lines 16-30, emphasis added.

As can be seen from the quoted passage, the interleave pattern is based on the unencrypted, one-time pad signal 24 which is generated by the random number generator 12 and not by replacing values in the key less than n by A and all other values in the key by B. Furthermore, what is interleaved in Cornelius et al. is the output of the pad encrypter 16 and one-time pad encoder 14, not values of A and B. Furthermore, now that the independent claims have been limited to redistribution of tasks for purposes of load balancing, it is respectfully submitted that a person of ordinary skill in the art could not reasonably be expected to look to art related to methods and apparatus for hybrid, one-time pad encryption and decryption apparatus with methods for encrypting and decrypting data. For those reasons, it is respectfully submitted that the obviousness rejection of claim 1 should be withdrawn.

In paragraph 17 of the Office action, the examiner acknowledges that neither the primary reference to Poeppelman nor the secondary reference to Cornelius et al. discloses "creating a key comprised of the reverse bit order of a serially indexed count from 0 to 2^z." The examiner relies upon a third reference to Stephen for the missing teaching citing column 3, lines 52-61. Citing more fully from column 3 provides the following:

> The two-stage interleaving process to temporally separate the input data stream includes a bit interleaving stage followed by a word interleaving stage. High data rate and efficient hardware can be achieved by using a small and fast memory for the bit interleaver and a large and slow memory for the word interleaver. FIG. 2A provides an illustrative example for the two-stage interleaving. In this illustrative example, the input data stream 210 consists of 12 data bits numbered 1, 2, 3, . . . 12.
>
> In the first stage of interleaving, the interleaving is performed on a bit basis. The input data stream is stored in a bit interleaver memory 215 in a bit normal order. Then the stored input data stream is read out in a bit scrambled order such that adjacent bits are separated. For example, the bit normal order is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12, while the bit scrambled order is 9, 2, 1, 5, 6, 10, 11, 4, 3, 7, 8, and 12.
>
> The scrambled bits are then converted into groups of N-bit words in a word grouping 220. In the example of FIG. 2A, N=2. The scrambled data bits are grouped into six groups of 2-bit words. The six groups are groups 221 (9,2), 222 (1, 5), 223 (6, 10), 224 (11, 4), 225 (3, 7), and 226 (8, 12).

> In the second stage of interleaving, the interleaving is performed on the word basis. The groups of N-bit words are then stored in a word interleaver memory 230 in a word normal order. Then, the stored words are read out in a word scrambled order such that adjacent words are separated. For example, the word normal order is (9, 2), (1, 5), (6, 10), (11, 4), (3, 7), and (8, 12), while the word scrambled order is (3, 7), (1, 5), (9, 2), (6, 10), (11, 4), and (8, 12).
>
> The scrambled words are then converted into an output data stream 240. The output data stream 240 is a serial data stream and includes 3, 7, 1, 5, 9, 2, 6, 10, 11, 4, 8, and 12.
>
> The word interleaver interleaves the words, while keeping the bits in each word together, thereby providing further separation of originally adjacent bits. This greater separation of adjacent bits allows greater error correction capability. Each stage of the two-stage interleaving process may be block or convolutional or any other type of interleaver. Column 3, lines 28-67.

The relevance of the cited two-stage interleaving process to the claim language is not apparent. This citation to a two-stage process can hardly be said to make up for the shortcomings of the primary and secondary references. For these reasons, it is respectfully submitted that the obviousness rejection of claim 1 should be withdrawn.

In paragraph 19 of the Office action, the examiner cites Poeppelman as teaching the invention of claim 2 substantially as claimed. The examiner cites column 3, lines 8-10 as teaching "selecting a portion of a list." The cited portion discloses a suffice generation step. The examiner cites column 3, lines 18-36 as teaching "renumbering the selected portion of the list to form a key." The cited portion is directed to the suffice generation step introduced earlier in the paragraph. In full, the section from Poeppelman is as follows:

> Referring to FIG. 3, an example of a suffix generation step (process) 300 is shown. Suffix generation 300 generally takes a small number of bits (e.g., 3) from a source word 302 and maps the bits into a larger number of bits (e.g., 4) of a code word 304. The term suffix generation is a general term used to describe the manipulation of only a few bits of the sequence (e.g., the last few bits of a large sequence). Since the upper few bits (e.g., most significant bits) are used in the example shown, the term suffix generation is used. Similar methods to those described herein may be implemented for more or fewer bits, other bits in the beginning of the source word (e.g., prefix or least significant bits) and/or any appropriate predetermined bits elsewhere in the source word 302. These predetermined (e.g., suffix, prefix, etc.) bits are generally implemented to break and/or prevent any predetermined constraint violation across code words 304 and to provide a unique code (e.g., mark) to indicate whether sequence replacement was implemented. In the 3-bit example for the 40-bit source word 302, 3 source bits are generally mapped (modified) into corresponding 4 code bits at the

respective (e.g., most significant bits) predetermined portion of the corresponding 41-bit code word 304 as follows:

Source	>>	Encoded
000	>>	0110
001	>>	1001
010	>>	0111
011	>>	1011
100	>>	1100
101	>>	1101

where a sequence replacement mark (SRM)=0011 may be implemented to indicate that a sequence replacement step (to be described in connection with FIG. 4) has been implemented. It is respectfully submitted that the primary reference's teachings relating to suffice generation utterly fails to "teach[ing] the invention substantially as claimed."

Note that the encoded suffixes in the code word 304 are generally designed to break any global or interleave constraint in the source words 302. Also note that each sequence replacement mark is generally unique (e.g., not part of the encoded suffix set). However, any appropriate source to encoded bit sequence and/or sequence replacement mark may be implemented to meet the design criteria of a particular application. Column 3, lines 6-49.

When the cited portions of the primary reference are closely examined, it is seen that the primary reference fails utterly in teaching "the invention substantially as claimed."

The assertion that Cornelius et al. discloses generating an interleave pattern in correspondence to a key and that Stephen discloses creating a list in which the entries are comprised of the reverse bit order of a serially indexed count from 0 to 2^z have been addressed above in connection with claim 1. Applicant reiterates the non-analogous art argument with respect to Poeppelman and Cornelius et al. made above in connection with claim 1. It is therefore applicant's position that the obviousness rejection of claim 2 should be withdrawn.

In paragraph 23 of the Office action, the examiner cites Poeppelman as teaching the invention of claim 6 substantially as claimed. The examiner cites column 3, lines 30-36 as teaching "creating a table

of interleave patterns for all values of lots of A and lots of B" and cites column 3, lines 40-45 as teaching storing the table. The cited portions are directed to the suffice generation step introduced earlier in the paragraph and cited in full above. It is respectfully submitted that when the cited portions of the primary reference are closely reviewed, the primary reference utterly fails in teaching "the invention substantially as claimed."

The assertion that Cornelius et al. discloses generating an interleave pattern in correspondence to a key and that Stephen discloses creating a list in which the entries are comprised of the reverse bit order of a serially indexed count from 0 to 2^z have been addressed above in connection with claim 1. Applicant reiterates the non-analogous art argument with respect to Poeppelman and Cornelius et al. made above in connection with claim 1. It is therefore applicant's position that the obviousness rejection of claim 6 should be withdrawn.

In paragraph 26 of the Office action, the examiner cites Poeppelman as teaching the invention of claim 9 substantially as claimed. The examiner cites column 3, lines 8-10 as teaching selecting a portion of a list, column 3, lines 18-36 as teaching renumbering the selected portion of the list to form a key, column 3, lines 30-36 as teaching creating a table of interleaved patterns for all values of lots A and lots B, and column 3, lines 40-45 as teaching storing the table. The cited portions are directed to the suffice generation step introduced earlier in the paragraph and cited in full above. It is respectfully submitted that upon close examination of the cited portions of the primary reference, the primary reference utterly fails in teaching "the invention substantially as claimed."

The assertion that Cornelius et al. discloses generating an interleave pattern in correspondence to a key and that Stephen discloses creating a list in which the entries are comprised of the reverse bit order of a serially indexed count from 0 to 2^z have been addressed above in connection with claim 1. Applicant reiterates the non-analogous art argument with respect to Poeppelman and Cornelius et al. made above in connection with claim 1. It is therefore applicant's position that the obviousness rejection of claim 9 should be withdrawn.

In paragraph 29 of the Office action, the rejection of claim 15 is the same as the rejection of claim 1. For the same reasons that claim 1 is believed to be in condition for allowance, claim 15 is believed to be in condition for allowance.

No arguments have been submitted in support of the patentability of the dependent claims. Applicant reserves the right to submit arguments in favor of the patentability of the dependent claims at a later date should that become necessary.

Applicant has made a diligent effort to place the pending claims in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested for pending claims 1-15. If the examiner is of the opinion that the instant claims are in condition for disposition other than through allowance, the examiner is respectfully requested to contact the undersigned attorney so that additional changes to the claims can be discussed.

Respectfully submitted,

Edward L. Pencoske
Reg. No. 29,688
Jones Day
One Mellon Center
500 Grant Street, Suite 3100
Pittsburgh, PA, USA, 15219
(412) 394-9531
(412) 394-7959 (Fax)
Attorneys for Applicant

Courtesy Copy filed with Amendment dated 27 Feb. 2008

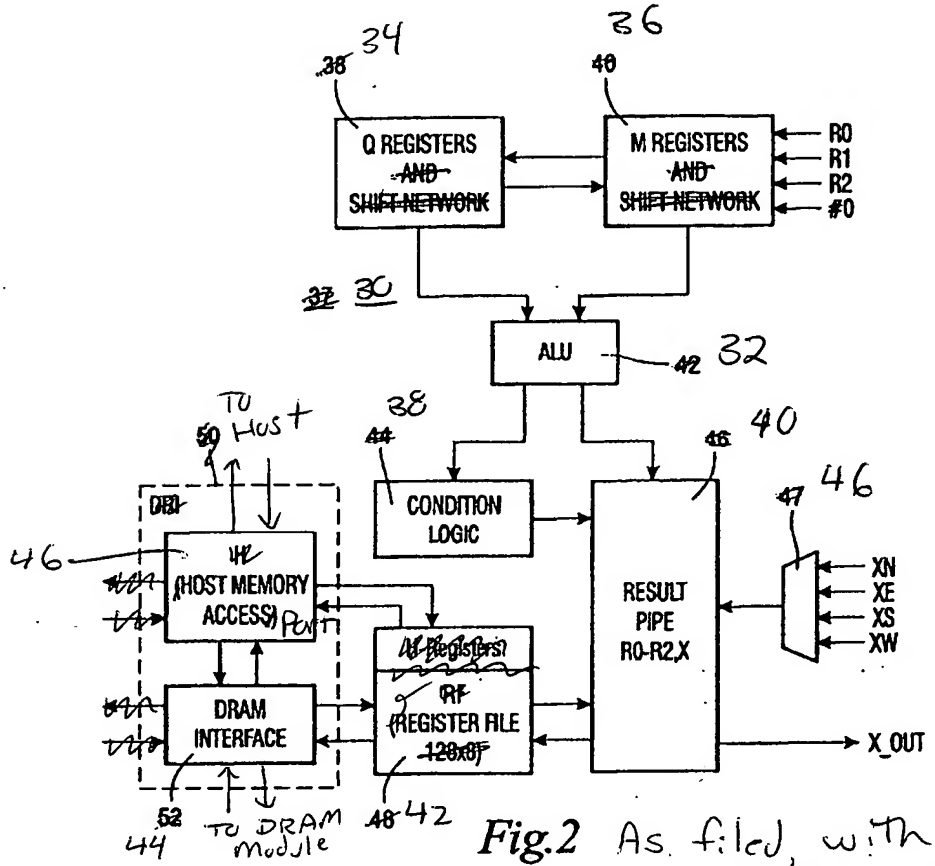

Fig.2 As filed, with changes marked in red to show differences between as filed FIG. 2 and FIG. 2 filed 23 Jan. 2004

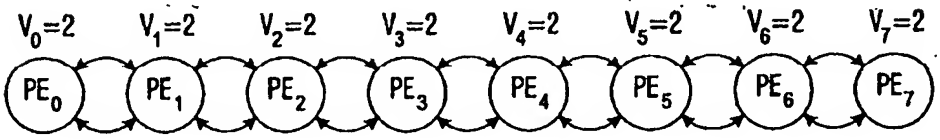

Fig.3

Annotated Sheet

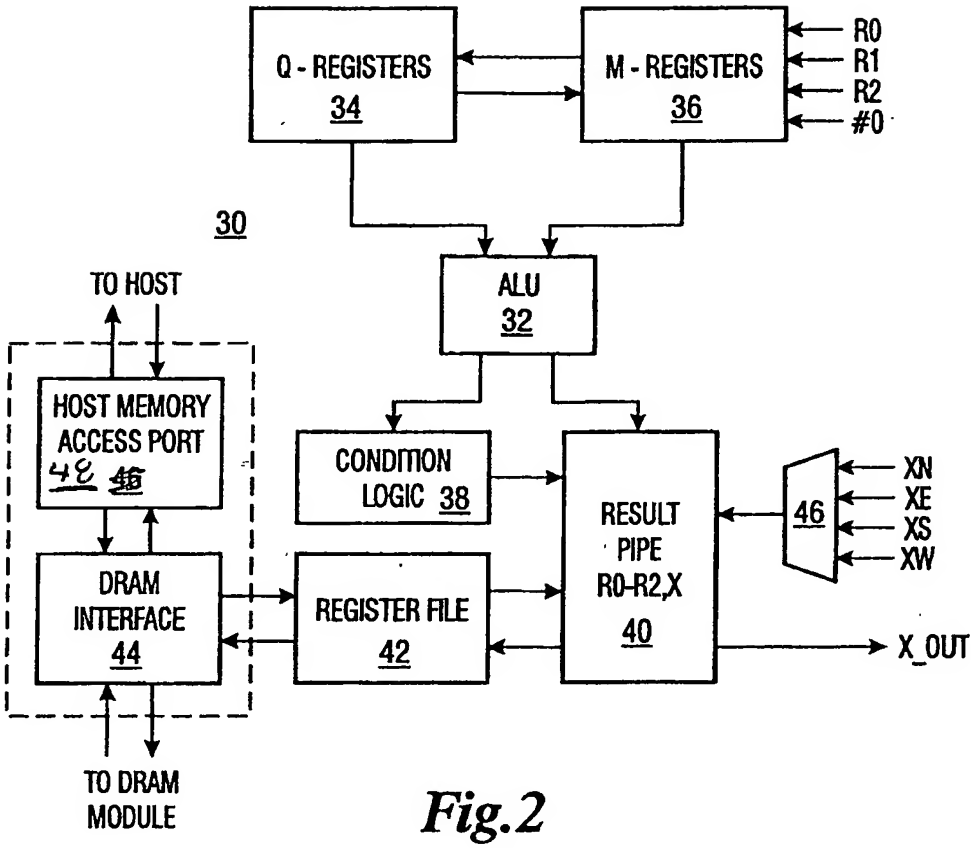

Fig.2

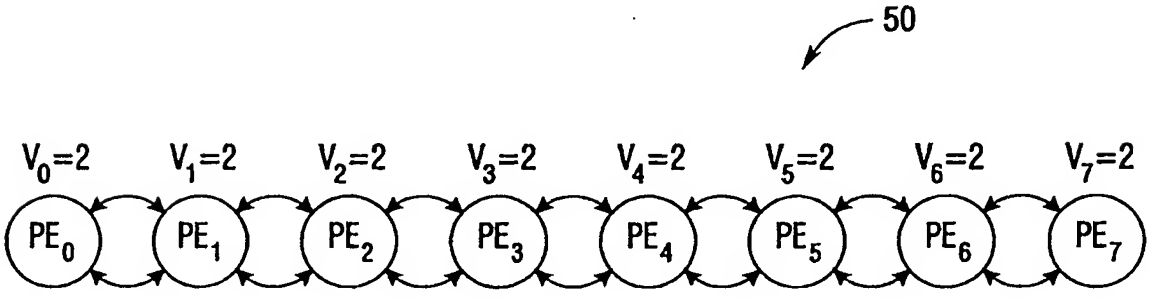

Fig.3